App Volumes provides a virtualized, real-time application delivery engine for virtual desktop infrastructure and is designed to enable VDI deployments to ensure greater flexibility, agility, and cost reduction.

This book starts with an in-depth overview of the architecture and components used to design an optimized solution. We then show you how to install and configure App Volumes for different use cases such as VMware View integration, using VMware ThinApp, Citrix XenApp, and more.

Throughout the chapters, we provide hints, tips, and tricks as well as best practices. By the end of the book, you will have built a working App Volumes environment and acquired the skills needed to build and run a production environment.

Who this book is written for
This book is for administrators and IT teams who would like to take full advantage of App Volumes to deal with application workloads. It will also be useful for virtualization professionals and teams who are working on VDI deployments and VMware end user computing solutions.

What you will learn from this book
- Find out how the VMware App Volumes solution can enhance the management and delivery of applications in your desktop environment
- Design a real-life App Volumes solution using best practices and following the recommended sizing guides
- Install, configure, and deploy App Volumes so it is ready to start delivering applications
- Create and prepare applications ready to be delivered to end users
- Discover how App Volumes can enhance other desktop solutions by looking at how it integrates with VMware Horizon View, VMware ThinApp, RDSH, and Citrix XenDesktop/XenApp
- Understand how to configure the advanced options within App Volumes
In this package, you will find:

- The author biography
- A preview chapter from the book, Chapter 1 'Introduction to App Volumes'
- A synopsis of the book’s content
- More information on Learning VMware App Volumes
Peter von Oven is an experienced technical consultant and has spent the past 20 years of his IT career working with customers and partners in designing technology solutions aimed at delivering true business value. During his career, Peter has been involved in numerous large-scale enterprise projects and deployments and has presented at key IT events, such as VMWorld, IP EXPO, and various VMUGs across the UK. He has also worked in senior presales roles and presales management roles for some of the giants of IT, such as Fujitsu, HP, Citrix, and VMware, and has been awarded VMware vExpert for 2015 and 2016.

Over the past 10 years and more, Peter has focused his skills and experience by specializing in the desktop and application virtualization market and today works for the market-leading desktop transformation specialists, Liquidware Labs.

Peter got his first taste for writing when assisting with some of the chapters in the book Building End-User Computing Solutions with VMware View, which then lead to four other Packt titles, VMware Horizon Mirage Essentials and VMware Horizon Workspace Essentials, co-written with Peter Bjork and Joel Lindberg, and VMware Horizon View Essentials and Mastering Horizon View, co-written with Barry Coombs.
When we talk about end-user computing, we are typically referring to the virtual desktop infrastructure (VDI), where desktop operating systems are virtualized and hosted centrally in a data center, and users connect remotely to them using some form of client endpoint device over an optimized network delivery protocol. But what about the applications?

Deploying VDI decouples the OS from the local hardware, making the OS portable. However, the applications still remain wedded to the OS, meaning you have to continually manage multiple OS images just to manage applications. After all, the only reason you have the OS in the first place is to run applications.

With the introduction of VMware App Volumes, you now have a solution that allows you to abstract applications from the OS of the virtual desktop and deliver them back individually and on demand to the end users' virtual desktop machine. This approach is now more commonly referred to as application layering. Application layering allows you to separate applications from the underlying OS and then deliver them back into the OS, merging the application files and the OS files together or layering the application in.

By doing this, you can not only manage applications independently of the OS, but also get one step closer to deploying the nirvanic solution of a truly stateless virtual desktop infrastructure.
What this book covers

Chapter 1, Introduction to App Volumes, gives you a high-level introduction to App Volumes. We talk about how it works and what benefits it can bring to your desktop environment. We also look at where it fits within the overall VMware EUC portfolio, where you would use App Volumes compared to some of the other application delivery/packaging technologies, before finally discussing some of the key use cases.

Chapter 2, Architectural and Feature Overview, takes you a little deeper into the different component parts that make up an App Volumes solution. Then, once you are familiar with these different components, we discuss how they fit and work together.

Chapter 3, Designing and Building an App Volumes Deployment, covers the recommendations for sizing, scalability, and making the solution ready for a production environment. The chapter starts by taking a step back and looking at what we have in our current environment, how we are going to approach the overall project, and how to move it forward. We look at both the business and technical elements of a project and break those down into three distinct project phases.

Chapter 4, Installing and Configuring the App Volumes Software, takes you through installing App Volumes and completing the initial configuration steps. Before we do that, we take a few minutes to walk through the prerequisites and what you need to have in place before you start the installation. We also cover the App Volumes infrastructure requirements.

Chapter 5, A Guided Tour of the Management Console, spends a short while walking you through an overview of the management console so that you can familiarize yourself with where the different options are located and how the console pages are laid out.

Chapter 6, Working with AppStacks, takes a deeper look into AppStacks, how they work, and how to start building them, before finally showing you how to assign them to end users. We also look at some of the best practices of how to build an AppStack. At the end of this chapter, you will be able to create, assign, and deliver an AppStack to an end user and then perform ongoing management tasks.

Chapter 7, Working withWritable Volumes, takes a much deeper look at the Writable Volumes feature of App Volumes, how they work, and how to create them, before finally demonstrating Writable Volumes in action. We also look at some of the best practices of how to build a Writable Volume, with particular attention to the dos and don'ts. At the end of this chapter, you will be able to create, assign, and deliver a Writable Volume to an end user as well as perform ongoing management tasks.
Chapter 8, Delivering ThinApp Packages with App Volumes, starts by discussing the background for the ThinApp solution and what it delivers, for those that have not used this solution before. We then go on to look at how and why you would deliver ThinApp packaged applications with App Volumes. Once we have an understanding of what ThinApp is and why you would use it, we use the Example Lab to create a ThinApp package and then build an AppStack that contains the ThinApp package. Once this is created, we go ahead and assign it to a user. In the final part of the chapter, we look at how you can use existing ThinApp packages in order to create AppStacks.

Chapter 9, Horizon View Integration, looks at how App Volumes integrates into the VMware Horizon View virtual desktop solution. We configure and build a VMware Horizon View deployment to deliver virtual desktop machines, with applications being delivered using App Volumes.

Chapter 10, Deploying App Volumes in a Citrix XenDesktop Environment, takes a closer look at how to deploy App Volumes to deliver just-in-time applications to a virtual desktop machine delivered by Citrix XenDesktop.

Chapter 11, Deploying App Volumes in a RemoteApp Environment, takes a closer look at how to deploy App Volumes to deliver just-in-time applications to a Microsoft RemoteApp environment. We build an environment and demonstrate how to deliver the AppStacks for hosted applications.

Chapter 12, Deploying App Volumes in a Citrix XenApp Environment, takes a closer look at the other key Citrix solution and how to deploy App Volumes to deliver applications that are published using Citrix XenApp.

Chapter 13, Deploying App Volumes in a Horizon View Hosted Apps Environment, looks at how to build and configure a Horizon View environment designed to deliver hosted applications to end users with the applications being configured as AppStacks and delivered using the Horizon View hosted app feature.

Chapter 14, Advanced Configuration and Other Options, is the final chapter and looks at some of the advanced configuration options, such as batch script files and App Volumes Agent configuration, as well as some other additional administrations tasks, such as how to customize the App Volumes templates used for creating AppStacks and Writable Volumes and how you can create your own template sizes.
Introduction to App Volumes

In this chapter, you will get a high-level introduction to App Volumes. We will talk about how it works and what benefits it can bring to your desktop environment.

We will also look at where it fits within the overall VMware End User Computing (EUC) portfolio, where you would use App Volumes compared to some of the other application delivery/packaging technology, before finally discussing some of its key use cases.

This book provides you with both the theory and the practical elements of App Volumes. Each chapter will explain specific areas of the solution and then give you the opportunity to try them out using an Example Lab. You will be introduced to the Example Lab at the end of this chapter.

What is App Volumes?

In August 2014, VMware acquired a start-up called CloudVolumes. The CloudVolumes technology provided a virtualized, real-time application-delivery engine for virtual desktop infrastructures as well as physical desktops.

In December 2014, CloudVolumes was rebranded and named App Volumes, and offered as a part of Horizon Enterprise Edition.

So, what does App Volumes give you? At a high-level, App Volumes provides a real-time application-delivery and life cycle management solution that is used as a delivery system for your virtual and physical desktops.
How does it work?

Let’s start with our typical virtual desktop environment. Although the desktop operating system has been abstracted from the underlying hardware, the applications are still tightly integrated into that operating system. The ideal virtual desktop solution is to deliver fully stateless desktops and have the user elements added on the fly, depending on the user’s profile. The fully stateless desktop model provides the most cost-effective solution by being easier to manage and requiring less infrastructure.

Today, there are a number of tools that can take care of delivering the user personalization, user data, and user profile elements to the desktop; however, applications are still delivered either as part of the base operating system image or using some form of application remoting or publishing.

App Volumes provides a layer of abstraction between the operating system and applications by delivering the applications in separate containers. These containers are called AppStacks and they integrate seamlessly into the operating system of the virtual desktop machine.

The following diagram shows the traditional, static model on the left and the App Volumes on-demand delivery method on the right:
As well as application containers, App Volumes provides the end users with their own container or virtual hard disk, into which they can install their own applications. This container, called a **Writable Volume**, "follows" them when they log in to different virtual desktop machines, bringing all their applications with them.

The following diagram illustrates the App Volumes model of having application containers (AppStacks) and user-writable containers (Writable Volumes):

So how do you get started? The first thing you do is create or capture an application that can be delivered by App Volumes. You start the process by installing the application on a virtual desktop machine, which is referred to as the provisioning machine. This provisioning machine is basically a vanilla installation of the operating system, with no applications installed.

When you start the capture process, an empty **Virtual Machine Disk** (VMDK) file, called an AppStack, is mounted on to the provisioning machine. Next, you start the application installation as you would normally do. All the files associated with this application are then redirected to the AppStack or VMDK file.

Once you have completed the capture process, the AppStack is set to read-only mode and is ready to be assigned to end users based on their **Active Directory** group membership. AppStacks can also be assigned to individual users or other groups.
For the users to be able to access the applications in their assigned AppStack, an App Volumes Agent runs on their desktop, mounting the VMDK file and making the application appear as if it were fully integrated and installed locally rather than running from an additional drive. This is how applications are able to be delivered to a user in real time, as AppStacks can be assigned on the fly.

As well as the applications that get delivered as part of the AppStack, a user is able to have their own disk or VMDK file on which they can install their own applications. This is called a Writable Volume and will also follow the user when they roam among virtual desktops, making it perfect for developers to have non-persistent virtual desktops.

In the next chapter, Chapter 2, Architectural and Feature Overview, we will dive a little deeper into what exactly each of the different components does, and then in Chapter 6, Working with AppStacks, and Chapter 7, Working with Writable Volumes, we will take a deeper dive into how App Volumes, AppStacks, and Writable Volumes work.

Why do you need App Volumes?
So the question is what App Volumes does differently and what benefits it will bring.

As we have already discussed, App Volumes gives you the ability to deliver applications to users in real time; however, you also benefit from being able to manage application life cycles much more easily, as well as a reduction in costs.

Once you have created your AppStack, it's simply a case of a few clicks to assign it to a user or group of users. Conversely, it's that simple to remove it as well. This means that, when a user requests a new application, for example, it's very easy to assign it to them without having to go and install it.

If you are running a virtual desktop environment, then App Volumes helps you drive towards achieving a truly stateless desktop since applications are immediately and dynamically made available upon login, or even while the user is already logged in.

A comparison between App Volumes, Mirage, ThinApp, and RDSH
A question that comes up most of the time when talking about App Volumes and application delivery in general, in a VMware context, is this: which technology should you use and when?
People often ask whether or not App Volumes is going to replace ThinApp and Mirage, and whether these technologies are still required. The key thing to point out here is that each solution addresses a specific use case, and they actually do not all do the same thing. In fact, combining all three delivers a more complete solution.

To the question of whether or not App Volumes is going to replace ThinApp and Mirage, the answer is no. All three technologies are key to the VMware vision and strategy, and won't be going anywhere for the foreseeable future.

In this section, we will discuss which solution you will need, why you will need it, and then see how they are complementary to each other.

Let's start with Mirage. **Mirage** is a centralized, Windows image-management solution, primarily designed to manage physical desktop PCs and laptops. It is also used to deliver the containerized desktop solution, **Horizon FLEX**, for delivering virtual desktops in a **BYOD** (short for **Bring Your Own Device**) environment on both Mac and Windows laptops.

Next up is ThinApp. **ThinApp** is an application virtualization/packaging technology that is primarily used when you need isolation between applications. For example, you might need to deploy an older version of an application that doesn't run on your current operating system version. You may also need to run multiple versions of applications; for example, you may need to run different versions of Internet Explorer. This is one of the key differences in what the solutions deliver. ThinApp is an application-packaging technology. Mirage and App Volumes do not provide any packaging capabilities.

We have outlined what these technologies deliver and we have talked about them being complementary. Depending on your use case, you can use them in combination.

When it comes to managing a physical desktop environment or delivering containerized desktops with Horizon FLEX, Mirage is the technology you should use. If you have applications that need to be isolated, you will combine them with ThinApp to create Mirage application layers, with the application layers containing those ThinApp packages.

If we now take virtual desktop environments and, in particular, those that are built using linked clones and have a floating user assignment, then App Volumes will be the ideal solution for this use case. When they log in, end users will be assigned a vanilla desktop from a pool of floating desktops. Their applications will then be deployed by simply mounting the relevant VMDK file, thus containing the applications in real time.
That was the model for non-persistent, linked, clone desktops, but what about virtual desktops that are built using full clones and have a persistent assignment? For this use case, all three solutions can be combined: Mirage to manage the operating system element, App Volumes to deliver the applications, and the applications potentially being packaged using ThinApp, if you require the isolation between applications.

The VMware documentation for the latest version of App Volumes talks about being able to deliver AppStacks to physical desktops. Although this is possible, Mirage would be a better option for managing physical desktops.

As you can see, there are several use cases that lend themselves to the different technologies, and this is the key to deciding which ones to use and when.

The following diagram shows a high-level overview of how the technologies complement each other and work together:

In the next section, we will discuss the key use cases for App Volumes in more detail.
App Volumes use cases
There are several key use cases for deploying App Volumes within your environment.

The first of these use cases is deploying App Volumes within a virtual desktop infrastructure environment.

VDI desktops
A virtual desktop infrastructure (VDI) is the primary use case for App Volumes. It allows IT administrators to deploy applications in real time and independent of the core operating system. The decision on who gets which applications is based on the user and group policies, depending on the user’s membership in Active Directory.

So why is this important and a key use case?
In order to deploy the most optimized VDI, you would want to opt for a floating, non-persistent model. This means that virtual desktop machines are delivered on demand from a pool of available machines and that nobody owns their own desktop. Everything is delivered on demand, and when the user logs off the virtual desktop machine gets destroyed, ready for the next user to log on.

This model would suit the majority of users; however, there are some users, such as developers, who would still require a persistent virtual desktop machine so that they have the ability to install their own applications that will remain when they log off rather than being destroyed when the virtual desktop machine they have been using gets destroyed. Deploying these types of virtual desktop machines is more costly and puts a greater load on infrastructure resources as you need more computing power and more storage than you would in a non-persistent deployment.

By deploying App Volumes, and the Writable Volumes feature in particular, users essentially get their own hard drive with their own user-installed applications on that drive. As they log on to a non-persistent, floating virtual desktop, their writable volume or hard drive gets attached and mounted to that virtual desktop machine. To the user, it looks as if their applications are natively installed on their desktop and that it is their own desktop.

The end users also benefit from having their core applications delivered immediately, and for the IT department, this means they can easily manage application delivery in real time, allowing them to assign new applications on the fly as well as deploy any updates to existing applications.
Allowing users to work with this model of virtual desktop machine deployment also means that the overall infrastructure requirements, particularly storage, are significantly lowered.

It's also important to note that we are not just talking about VMware virtual desktop solutions. App Volumes also supports Citrix environments, and can be purchased specifically for this, which we will cover in the How to license App Volumes section of this chapter.

**RDSH-delivered applications**

This particular use case is one that often gets overlooked since Remote Desktop Services Host (RDSH) is in itself an application-delivery mechanism. So how can App Volumes help with this use case?

If you look at the basic infrastructure around RDSH, applications have to be installed onto the RDSH server before they can be published to the end users. This in itself makes the infrastructure pretty static in nature. By static we mean that if you need to scale up your RDSH deployment, you not only have to build a new server but also install the applications all over again afterwards. It's a similar story when you have to update applications.

App Volumes, in this use case, works in the same way it would with a VDI, but instead of mounting AppStacks on the virtual desktop machine's operating system this time you are mounting them on the RDSH server. It's just a case of installing the App Volumes Agent and capturing an AppStack for use in this environment. Obviously, your RDSH servers need to be running as virtual machines.

Essentially, you are creating a stateless RDSH server that allows you to add additional server resources and just mount the AppStacks on them rather than having to install the applications each time.

**Citrix XenDesktop and XenApp environments**

We already mentioned that App Volumes can be deployed in a Citrix environment and can deliver exactly the same benefits. XenDesktop-based virtual desktop machines work in exactly the same way as a VMware Horizon View virtual desktop machine when it comes to delivering applications; however, you can now support the Virtual Hard Disk (VHD) format.
**XenApp** works using the functionality of an RDSH server. This means that you can now deploy stateless XenApp Servers by mounting AppStacks to RDSH servers, as we discussed previously. Like XenDesktop deployments, this can also be done using VHD deployments if you use a virtualization platform based on XenServer or Hyper-V.

**Project A$^2$**  
Announced at VMworld in 2015 as a tech preview, Project A$^2$ brings together AirWatch enterprise mobile management (EMM) and VMware App Volumes to deliver mobile-like management for their Microsoft Windows 10 devices and applications.

**How to license App Volumes**  
There are several ways in which you can purchase and license App Volumes.

Firstly, App Volumes is a standard component of the Horizon Enterprise Edition license.

The second option is to purchase App Volumes as a part of the VMware Horizon Application Management Bundle. This bundle includes all the components except the core VDI solution, Horizon View, and is designed to complement existing Citrix environments.

Lastly, you can purchase App Volumes as a standalone product as part of an *à la carte* list of solutions. Whichever option you choose to purchase App Volumes, the product itself is licensed on a per-concurrent-user basis.

**How does this book work?**  
Now that we have covered the introduction to App Volumes, let's just take five minutes to introduce you to how this book works and what you will need to get the most out of it.

This book will guide you through the key use cases for App Volumes, from the theory of how it works and how to design a solution, to the more practical elements, such as installation.

Each section of this book has been written based on everyday, real-world experiences of working with the product in both production and proof-of-concept/pilot environments.
In parallel with the chapters, you can set up your own lab environment, which will cover practical examples of how the technology works, giving you the option to follow the steps and try them for yourself. This can either be as an exercise while reading the chapters or to assist you in setting up a proof-of-concept, pilot, or live deployment.

**Introducing the Example Lab**

Accompanying this book, as previously mentioned, is an Example Lab that you can build along the way. In this introductory section, we are going to cover the requirements for this lab and what it will look like. The lab is optional, and you can choose to build your own environment.

**What you will need for the lab**

The lab environment used in this book was built on a Dell PowerEdge R720xd LE configured with two E5-2680V2 2.8-GHz 10-core processors and 64 GB RAM, running VMware vSphere 6 as the hypervisor.

Although the server had internal storage, the lab was built using a Tintri all-flash array. By using Tintri and its VM-aware capabilities, you can monitor right down to the virtual-machine level.

The reason I chose Tintri storage is the fact that it has been developed from the ground up for virtualized environments removing contention points and collision domains such as the LUN construct. Tintri was very easy to deploy. It was done in just under 15 minutes. The folks at Tintri have created a new storage category. They call it **VAS** (short for **VM Aware Storage**). This "awareness" is backed by Tintri’s analytics engine learning the behavior of individual VMs underpinned by an intelligent filesystem. The architecture is designed to adapt to the fluidity that comes with virtualized workloads enabling it to deliver the right quality of service where and when it is needed. This makes Tintri ideal for VDI-type deployments as it has the ability to respond to performance bursts that can result from use cases such as App Volumes.
All of this was used to host the following virtual machines. The computer names of the virtual machines are shown in brackets and will be referred to in the lab examples throughout the book:

- Windows Server 2012 R2: configured with AD, DNS, and a file server (DC)
- VMware vCenter Server Appliance (VCSA)
- VMware ESXi v5.5 host server (ESX-1)
- Windows Server 2012 R2: App Volumes Manager (APP-VOL-MGR)
- Windows Server 2012 R2: Microsoft SQL 2008 Express*
- Windows Server 2012 R2: configured with the RDSH role**
- Windows Server 2012 R2: Horizon View Connection Server (VIEW-CS)***
- Windows 7: AppStack-provisioning desktop (AV-PROVISION)
- Windows 7: used as a virtual desktop machine (WIN7-DESKTOP-1)
- Windows 7: used as a virtual desktop machine (WIN7-DESKTOP-2)
- Windows 7: virtual desktop machine delivered by Horizon View (VIEW-1)
- Windows XP: used for ThinApp Setup Capture (WINXP-DESKTOP-1)

* In the Example Lab, the Microsoft SQL Express server is also installed on the same server as the App Volumes Manager. As this lab is just for testing, it is fine to configure it in this way. This is not recommended for production environments.

** Only required in Chapter 11, Deploying App Volumes in a RemoteApp Environment. If you don't plan to install and test this feature, there is no need to deploy an RDSH server.

*** Only required in Chapter 9, Horizon View Integration. If you don't plan to install and test Horizon View, there is no need to deploy a Horizon View connection server.

If you plan to build a lab as you follow the chapters, it's worth building the infrastructure detailed here beforehand. Build and install all the virtual machines, but with the exception of installing any of the App Volumes software components.

In writing this book, VMware vSphere 6 has been used as the hypervisor to host all of the VMs used in the Example Lab, but you could also run this in VMware Workstation.

If you choose not to follow the steps described, they will still help by providing you with a reference and foundation for building and managing your own environment.
Introduction to App Volumes

A schematic diagram of the core lab environment used for the examples is detailed in the following diagram:

The next schematic diagram shows the optional lab components for integrating App Volumes with VMware Horizon View and RDSH servers:
The file server in the lab environment was used to host a shared folder containing all the installation files so that they are accessible by all the other virtual machines.

This final schematic diagram shows the optional lab for Chapter 8, Delivering ThinApp Packages with App Volumes:

We also require an administrator account and a number of Organizational Unit (OU) groups and user accounts.

For administering App Volumes, a group called AppVol Admin was created. The administrator account was then added to this group. You can, of course, create other App Volumes administrator accounts.
In addition to the administrator account, we have created a number of Active Directory groups to reflect different departments. In each group, there are a few user accounts. These have been set up as shown in the following table:

<table>
<thead>
<tr>
<th>Active Directory Group 1</th>
<th>Active Directory Group 2</th>
<th>User</th>
<th>Login Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain Users</td>
<td>Sales</td>
<td>John Smith</td>
<td><a href="mailto:jsmith@pvolab.com">jsmith@pvolab.com</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sue Jones</td>
<td><a href="mailto:sjones@pvolab.com">sjones@pvolab.com</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bob Cooper</td>
<td><a href="mailto:bcooper@pvolab.com">bcooper@pvolab.com</a></td>
</tr>
<tr>
<td></td>
<td>Engineering</td>
<td>Peter Owens</td>
<td><a href="mailto:powens@pvolab.com">powens@pvolab.com</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td>James Thomas</td>
<td><a href="mailto:jthomas@pvolab.com">jthomas@pvolab.com</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ellie Richards</td>
<td><a href="mailto:erichards@pvolab.com">erichards@pvolab.com</a></td>
</tr>
<tr>
<td></td>
<td>Finance</td>
<td>Tom Jackson</td>
<td><a href="mailto:tjackson@pvolab.com">tjackson@pvolab.com</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Charlotte Edwards</td>
<td><a href="mailto:cedwards@pvolab.com">cedwards@pvolab.com</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Nick Mason</td>
<td><a href="mailto:nmason@pvolab.com">nmason@pvolab.com</a></td>
</tr>
</tbody>
</table>

These users and groups will be used throughout the Example Lab to demonstrate user assignments.

**Summary**

In this chapter, we have covered a high-level introduction to App Volumes, looking at what it is, how it works, and what it can deliver.

We then went on to cover exactly where App Volumes fits into the overall VMware EUC portfolio, bearing in mind that products such as ThinApp and Mirage seem to deliver almost the same functionality.

Finally, we saw how to get the most out of this book, and you were introduced to the Example Lab that you can build along the way in each chapter.

In the next chapter, *Chapter 2, Architectural and Feature Overview*, we will start to look a little deeper into each of the App Volumes components.
Where to buy this book

You can buy Learning VMware App Volumes from the Packt Publishing website.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

Click here for ordering and shipping details.